REMARKS

I. Summary of the Examiner's Action

A. Claim Rejections

As set forth in paragraph 4 on page 2 of the March 15 Office Action, claims 1 – 15 stand rejected under 35 U.S.C. § 101 as being directed non-statutory subject matter.

As set forth in paragraph 5 on page 5 of the March 15 Office Action, claims 1, 3, 8, 10 and 15 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over "A Neural Network Modelling Methodology for the Detection of High-Risk Programs" by Khoshgoftaar *et al.* (hereinafter referred to as the "Khoshgoftaar1 reference") in view of "Application of Neural Networks to Software Quality Modeling of a Very Large Telecommunications System" by Khoshgoftaar *et al.* (hereinafter referred to as the "Khoshgoftaar2 reference").

As set forth in paragraph 7 on page 12 of the March 15 Office Action, claims 2, 4 – 7, 9 and 11 – 14 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the Khoshgoftaar1 and Khoshgoftaar2 references as applied to claims 1, 3, 8, 10 and 15 and further in view of "Self Organizing Maps as a Tool for Software Analysis" by Pedrycz *et al.* (hereinafter referred to as the "Pedrycz reference").

II. Applicant's Response – Claim Rejections

A. Rejection of Claims 1 – 15 under 35 U.S.C. § 101

Applicant respectfully submits that it is not seen how the relied-upon portions of the MPEP (2106(IV)) are relevant to the claims of this application. The claims on their face are clearly statutory, falling into several of the four enumerated categories. The Examiner apparently is arguing that the claims, although *prima facie* statutory, are in some way directed to an abstract idea, law of nature, or a natural phenomenon. As such, the invention must produce a concrete, useful and tangible result to remain statutory.

In applying the legal standards at pages 2 – 4 of the March 15 Office Action, the Examiner focused on the concrete, useful and tangible result test. However, there are no factual findings presented on why this test should even be applicable. There must be a finding that the claims are directed in some way to an abstract idea, law of nature, or a natural phenomenon before the concrete, useful and tangible test is even applied. The Examiner merely assumed that the claims are directed to an abstract idea, law of nature or a natural phenomenon. The MPEP at 2106(C) states the following (emphasis added):

> While abstract ideas, natural phenomena, and laws of nature are not eligible for patenting, methods and products employing abstract ideas, natural phenomena, and laws of nature to perform a real-world function may well be. In evaluating whether a claim meets the requirements of section 101, the claim must be considered as a whole to determine whether it is for a particular application of an abstract idea, natural phenomenon, or a law of nature, and not for the abstract idea, natural phenomenon, or law of nature itself.

Such an evaluation is necessary because precedent is not applicable unless there is congruence between the facts of the case establishing the precedent and the case is in issue.

The Background section of the application goes into great detail concerning the genesis of the invention, part of which is reproduced later in this response. A focus of the invention is the ability to determine when a data structure used by computer program is "risky". A data structure used by a program may become risky when the computer program is executed on a computer platform different from the computer platform for which the computer program was designed. Since the so-called "porting" of code to new platforms is a commonplace in the field, it is inexplicable how the claimed invention could possibly be found not useful. It would seem every that time an application program is ported to a new computer environment that the present invention would find use.

Although Applicant submits that the claims do meet the useful, tangible and concrete result test, Applicant respectfully argues that it is clear error to even apply the test since the claims are not directed to a judicial exception where claims that are *prima facie* statutory are denied patentability because they are directed in some way to an abstract idea, law of nature or natural phenomenon.

Regarding the useful, tangible and concrete result test, Applicant submits the following. Clearly the ability to determine whether data structures associated with a

computer program are risky is very useful. In addition, as set forth at MPEP 2106C2(2)(b), the opposite of "tangible" is "abstract". Having a real world idea of whether a data structure associated with a computer program will cause an error at run time is not some "abstract" characterization of the nature of a data structure, but a highly practical characterization of the data structure. Further, regarding "concrete", as set forth at MPEP 2106C2(2)(c), the opposite of "concrete" is unrepeatable. There is no finding in Examiner's rejection on why the subject matter of the claims would function in an unpredictable and unrepeatable way. Accordingly, even assuming that the useful, tangible and concrete result test is applicable (which the Applicant admits only for the sake of argument), the claims are still statutory.

In view of the foregoing, Applicant respectfully requests that the rejection of claims 1 – 15 under 35 U.S.C. § 101 be withdrawn on this basis.

B. Rejection of Claims 1, 3, 8, 10 and 15 under 35 U.S.C. § 103(a)

Applicant reproduces claim 1 (as amended) here as a convenience to the Examiner:

> 1. A method comprising:
>
> using a neural network to detect degrees of risk of at least one data structure established by a computer program code, the data structure comprising at least two data elements and the neural network comprising at least two neurons, the neurons being related to each other by a topological arrangement involving a neighborhood definition, each of

the neurons comprising a vector for representing elements of an input data space, at least one neuron having an associated label indicating the type of the neuron, extracting information concerning the at least two data elements from the at least one data structure,

forming at least two input vectors from said extracted information of the data elements, the vectors being compatible with the vectors of the neurons,

comparing said input vectors with said vectors of the neurons,

detecting a type of said at least one data structure by using an associated label obtained on the basis of said comparison, and

presenting the type of said at least one data structure as data indicating a degree of risk of said at least one data structure.

Applicant respectfully submits that it is not seen where the references of record, whether taken singly or in combination, either describe or suggest the subject matter of claim 1.

In particular, although Applicant's invention and the relied-upon primary references involve the use of neural networks to measure program reliability; they are directed to different problems within this topic. The Khoshgoftaar references concern reliability issues that generally arise from program complexity. Applicant's invention, on the other hand, concerns reliability issues associated with data structures. Reliability issues created by data structures may arise from re-using pre-existing code on, for example, a second platform when the code was designed for use on a different first

platform. The data structures established when the pre-existing code is executed on the second platform may be fault-prone in a way not encountered when the pre-existing code was executed on the first platform. These problems are explained at length in the Application at page 1, line 9 – page 3, line 24 (emphasis added):

> "Runtime errors of applications compiled to different operating system platforms are often affected because of incorrect function of memory reservation of the application. There may be restrictions in the operating system or architecture, for example, from what address may that space begin that is reserved to one parameter or variable or how large is the size of the memory unit to be reserved to the parameter or variable. Sometimes, a clear error is not created and the execution of the application continues, but the memory space of the application gets corrupted because of an incorrect data structure. There are some heuristic rules for checking data structures from computer program code, but they are usually rather laborious to perform.
>
> * * *
>
> Common hazardous use of data types is caused by improper pointing and casting of the data types belonging to a data structure, which occurs especially easily in the case of arrays. The result of an operation where the items of the array of s2 are referred to via direct memory operation like incrementing an address variable so that it points to the 4th short int element of the array defined in the example varies depending on the operating system of the compiler. The content of the address could be the wanted A[1].a1 of the padding bytes 123 and 124. The similar situation occurs in the structure 's1' case, wherein the content of the bits 8 and 9 could be the wanted 'c1' (106) of the padding bytes (115 and 116) varying depending on the operating system compiler.

> It is difficult to detect problems related to memory structure mapping when reusing software from devices of different processor and memory mapping systems or when a different compiler is used."

Applicant has amended the claims to improve their overall quality, and in the process of so doing, have emphasized that Applicant's invention is directed to using a neural network to assess degrees of risk of data structures associated with computer programs. Establishment of a particular data structure, as described by the application, may result from in non-limiting examples compilation of computer program code by a particular compiler; execution of computer program code on a particular computer platform; or by definition in the computer program code itself.

That the Khoshgoftaar1 reference, for example, is concerned with using neural networks to gauge program reliability based on program complexity is shown by this portion appearing on page 304 (emphasis added):

> "Recent advances in artificial neural network programming techniques have attracted the attention of software engineers. Researchers have applied neural networks to several applications related to software quality control. Karunathi *et al.* developed neural network models of software reliability growth, and demonstrated that these models have better predictive quality than some analytic models [4]. Khoshgoftaar *et al.* proposed a neural network approach for predicting the number of faults in program modules [6]. This article focuses on feed-forward neural network classifiers for static patterns with continuous-valued inputs. The input patterns are the software complexity metrics associated with

program modules. The neural network classifies the module as either high risk or low risk."

Since the Khoshgoftaar references show no appreciation for using neural networks to assess degrees of risk of data structures established by computer program code, Applicant respectfully submits that claim 1 and the remaining independent claims are patentable over the art of record, whether taken singly or in combination. Applicant therefore respectfully requests that the rejection of claims 1, 8 and 15 be withdrawn. Applicant also respectfully requests that the rejection of dependent claims 3 and 10 be withdrawn both since these claims depend from allowable base claims and for reasons having to do with their independently-recited features.

C. Rejection of Claims 2, 4 – 7, 9 and 11 - 14 under 35 U.S.C. § 103(a)

Applicant respectfully submits that it is not seen where the Pedrycz reference remedies the deficiencies of the Khoshgoftaar1 and Khoshgoftaar2 references. As claims 2, 4 – 7, 9 and 11 – 14 depend either directly or indirectly from independent claims that were distinguished over the combination of the Khoshgoftaar1 and Khoshgoftaar2 references, Applicant respectfully submits that claims 2, 4 – 7, 9 and 11 – 14 are likewise patentable. Applicant therefore respectfully requests that the rejection of claims 2, 4 – 7, 9 and 11 – 14 be withdrawn as well.

III. Conclusion

The Applicant submits that in light of the foregoing remarks the application is now in condition for allowance. Applicant therefore respectfully requests that the outstanding rejections be withdrawn and that the case be passed to issuance.

Respectfully submitted,

May 14, 2007 — Date

David M. O'Neill (35,304)

David M. O'Neill (35,304)
Customer No.: 29683
HARRINGTON & SMITH, LLP
4 Research Drive
Shelton, CT 06484-6212
Telephone: (203) 925-9400
Facsimile: (203) 944-0245
Email: DOneill@hspatent.com

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. 1450, Alexandria, VA 22313-1450 on the date indicated.

May 14, 2007 — Date

Name of Person Making Deposit